

Third-Person Navigation of Whole-Planet Terrain in a Head-tracked Stereoscopic Environment

Zachary Wartell, William Ribarsky and Larry Hodges
GVU Center, Georgia Institute of Technology

Abstract

Navigation and interaction in virtual environments that use stereoscopic head-tracked displays and have very large data sets present several challenges beyond those encountered with smaller data sets and simpler displays. First, zooming by approaching or retreating from a target must be augmented by integrating scale as a seventh degree of freedom. Second, in order to maintain good stereoscopic imagery, the interface must: maintain stereo image pairs that the user perceives as a single 3D image, minimize loss of perceived depth since stereoscopic imagery cannot properly occlude the screen's frame, provide maximum depth information, and place objects at distances where they are best manipulated. Finally, the navigation interface must work when the environment is displayed at any scale. This paper addresses these problems for god's-eye-view or third person navigation of a specific large-scale virtual environment: a high-resolution terrain database covering an entire planet.

1. Introduction

Navigation through extended virtual spaces is a topic that has received significant attention over the years. However, there has been relatively little work specific to navigation in stereoscopic HTD's (head-tracked displays). HTD's are distinguished from HMD's (head-mounted-displays) in that HTD's do not attach the display surface to the user's head. Examples are the CAVE, fish tank VR and the virtual workbench. Interaction and viewing in stereoscopic HTD's present problems that do not come up in simpler environments. These include: the maintenance of good stereo effect as one moves closer or farther away from objects (especially if the change of scale is great); the need to handle some degrees of freedom differently than in monoscopic interfaces; the problems with stereo fusion at certain viewing positions; the collapsing of the stereo effect due to clipping by the screen boundary or occlusion by user hands or bodies; and the difficulties with accurate selection and detailed manipulation caused by the simultaneous stereo images. In addition there are problems with handling up to seven degrees of freedom

simultaneously, continuously, and easily. We will address these problems for third person navigation on a single screen HTD, the virtual workbench.

2. Challenges

Navigating through large scale virtual spaces such as whole-planet terrain on stereoscopic HTD's presents three major challenges:

- managing 7 degrees of freedom: the 6 Euclidean ones plus scale
- maintaining good stereo imagery
- ensuring that navigation methods work at all spatial scales

The necessity of supporting 7 degrees of freedom may not be apparent to those more familiar with simpler VR input and output devices. For example, take an application running on a regular monitor. The user can get a global view by simply moving the viewpoint away from the scene or she can get an arbitrarily detailed view by moving the viewpoint closer to an object. (While physically impossible, this works in interactive graphics due to the pin-hole camera model.) In this example, zooming is accomplished by viewpoint movement. When a VR interface employs a head-tracked display, a stereoscopic display, or direct manipulation with a six DOF device, however, this "zooming" by translation must be augmented with zooming by scaling. With head-tracking, zooming out by moving the viewer away from an object will rapidly increase the sensitivity of the projected image to head-position (Figure 1a). Such sensitivity can be quite distracting. By using an independent scale dimension, the system can scale down the object and preserve the object-viewer distance in order to avoid this problem. In the stereoscopic case, zooming by moving the viewer towards or away from an object can bring the object either far above the display surface or far below the display surface (Figure 1b). In both cases the resulting screen parallax is likely to grow too large, causing the user to see double images (diplopia). (Screen parallax is the distance on the screen between the projected positions of a point as seen by the right and left eye.) Scaling the object while keeping the object near the projection plane solves the problem. Finally direct manipulation using a six DOF device will be difficult for large objects. Manipulating objects is

easiest when the user can see the complete object and the object is within arms reach of the viewer [19]. If, in order to see the complete object, the viewer must move away from the object, both of these requirements cannot be satisfied (Figure 1c). Again the solution is to scale so that the object is small enough to be brought close to the user and still be viewed in its entirety. Note that properly implementing this scaling differs from field-of-view manipulation, since in typical VR systems, the FOV should be determined by the display's physical configuration and not by some arbitrary parameter.

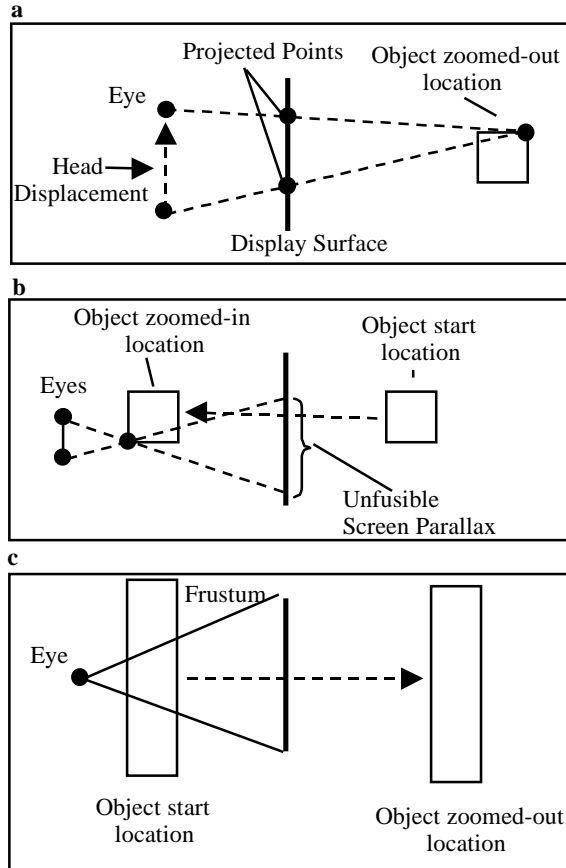


Figure 1: Stereoscopic zooming

The second challenge in large scale stereoscopic navigation is maintaining good stereo imagery. This requires balancing four goals. First the screen parallax of the stereo images should yield retinal disparities that are within fusible limits of the user. We use Yeh and Silverstein's [27] measured vergence tolerances of 4.93 degrees of cross disparity for objects in front of the display surface and 1.57 degrees of uncrossed disparity for objects behind the display surface. Second the system should provide a good sense of stereoscopic depth. This can be maintained by various distortions and/or keeping the stereo imagery close to the viewer where stereo is a strong depth cue [5]. Thirdly, as previously mentioned, bringing the stereo imagery within arms' reach improves

direct manipulation. In general, this also entails keeping the manipulated objects above the display surface where the user can contact them. Finally, the navigation method should avoid losing the illusion of depth due to frame cancellation [29]. This occurs when a continuous surface extends above the display surface and beyond its edges. As the user moves his head, different parts of this surface are clipped by the sides of the view frustum. Since occlusion is one of the strongest depth cues, this clipping often cancels the depth illusion.

The final challenge is ensuring that the navigation technique works at all scales. For example in terrain navigation, at one extreme a planet can look like a globe floating on the display while at the other extreme terrain features appear at their true size. Since research suggests that it is necessary to provide both first person and third person view points and corresponding modes of navigation [16][7], one must ensure each mode individually works at all scales.

3. Previous work

Much literature exists concerning navigating large information spaces such as [15][3][2][8]. However, this body of work concerns 2D GUI interfaces. 3D work for large spaces, exemplified by the April issue of *Presence: Teleoperators and Virtual Environments* [1], has focused on psychological studies, not specific techniques.

Many general navigation techniques for smaller scale virtual environments have been developed, studied and compared. Much of this work, however, concerns first person travel techniques [4]. There is much less published on third person travel techniques. A few relevant examples are pre-screen projection [12], Netra [10] and WIM (Worlds In Miniature) [25]. While pre-screen projection is used for map navigation, the technique is controlled by head movements which conflict with the use of head-tracking and stereoscopic display on a virtual workbench. In Netra the user manipulates the orientation of virtual MRI of a patient's head with a tracked doll's head. However, while similarly manipulating a planet would work at small scales this method would not work well at large scales. More importantly for our goals, Netra uses neither head-tracking nor stereoscopic display. Finally, the WIM technique displays a small copy of the world which the user holds. The user can both manipulate objects and travel using the WIM model. We are unaware, however, of WIM extensions to the workbench that incorporate zooming. Furthermore, even with such an extension, a basic travel method addressing the challenges of section 2 would still be needed.

Head-tracked stereoscopic displays have been studied for several years [18][6][28] and more recently, the virtual workbench has gained much attention [9][14][20][21][22][11][7]. However, this body of work discusses applications with relatively small scale models

such as human organs, cars, molecules and local terrain. More importantly, these papers do not address in detail the challenges listed in section 2.

There is a small set of work which addresses stereoscopic terrain display and navigation. Some of the earliest work [26] avoids diplopia in the following manner. The authors first fix the distance between the viewer and display surface so that objects at infinite distances beyond the projection plane remain fusible. Next they fix the near clipping plane to avoid rendering objects which are too close to the viewer to be fusible. However, they do not use head-tracking nor six DOF control devices. Also they do not address scale issues nor whole-planet terrain.

Ware and co-workers refine the display of non-headtracked, stereoscopic terrain [29]. Their methods are based on the premise that correct stereo is not always the best choice. They first scale the world about the central eye point to bring the nearest visible terrain point to the screen. Then they adjust the modeled eye separation to enhance stereoscopic depth based on the nearest and farthest sampled pixel in the Z buffer. Unfortunately, for the head-tracked case, false-eye separation modeling is problematic. It induces shearing of the stereo image from head movement [30]. Finally, Ware's environment does not address six DOF devices or head-tracking and does not support very large scale virtual spaces.

More recently Durbin et. al. reported on a command and control application called Dragon on the virtual workbench [7]. Dragon presents terrain and military icons on the virtual workbench. One of the authors of our paper designed and implemented the head-tracked stereo software and helped develop the navigation system for Dragon. Navigation in Dragon addresses some, but not all, of the issues from section 2. Dragon does treat scale as a separate degree of freedom and couples user scale and user location in the map-centric (third person) navigation method. However, Dragon's navigation does not explicitly address the stereo issues we raised in section 2.

4. Display environment

Our environment consists of a Fakespace Immersive Workbench with Polhemus trackers. The user wears stereo Crystal Eyes glasses with a tracker receiver mounted on the side. The user holds a custom-made "laser pointer"-like device which is tracked and has 5 buttons. The physical device appears to emit a virtual laser beam.

It is useful to have a naming convention in the viewing coordinate system hierarchy. Ours is similar to that of [24]. In our scheme (Figure 2), the top coordinate system is the platform coordinate system. This is the coordinate system which is manipulated to move and scale the user. Directly attached to that coordinate system is the projection plane coordinate system and the

emitter coordinate system. The projection plane coordinate system contains the projection plane in the XY plane with the window centered about the origin. The emitter coordinate system simply represents the tracker emitter. Attached to the emitter coordinate system is the head receiver coordinate system and attached to that is the eyes coordinate system. The two eye points are on the x-axis of the eyes coordinate system and are symmetric about the origin.

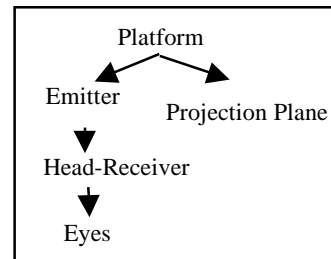


Figure 2: Viewing coordinate hierarchy

5. Third person navigation

With this display setup in mind, we now discuss the third person navigation method, navigating from a god's-eye or map-centric point of view. First we cover the start position and then we discuss the three navigation modes.

5.1. Start position

Any 2D or 3D map navigation software must obviously have an initial point of view. For 2D maps and small scale 3D maps, the start point is trivially the view point that displays the entire map. When navigating whole-planet terrain in a head-tracked stereoscopic display, choosing a good starting point is more complex. From our experimentation, we characterize an ideal start position as follows:

- The position should display the most complete view of the planet possible.
- The user should be able to view the entire planet without diplopia or frame cancellation.
- The user should be able to physically reach as much of the displayed planet as possible directly with his hands.
- Within the previous three constraints, the planet should appear as large as possible.

We choose a start position that empirically balances these goals. We parameterize this start position based on:

- the display size and position
- a standard height (SH) measuring an average user's eye height
- a default viewer distance (D) from the front of the workbench (Figure 3).

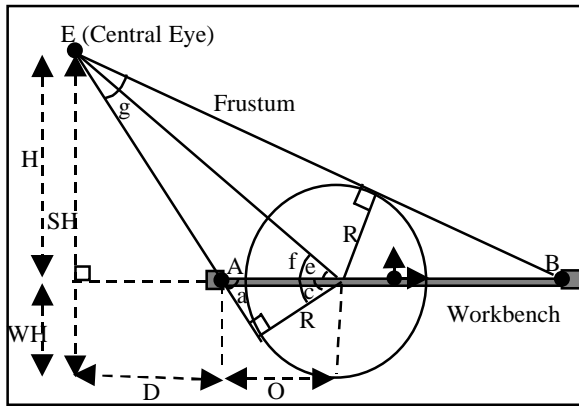


Figure 3: Start position

We compute the radius, R , and offset, O , of the largest sphere which is contained in the default view volume. Figure 3 shows a side view of the situation. We fix the sphere center to be in the projection plane in order to keep half of the planet above the physical display. While we could place more of the planet above the display, this would require using a smaller scale planet.

In Figure 3, WH is the height of the display surface. Points A and B represent the edges of the display area. Lower case letters represent the illustrated angles. Computation is then done in a 2D coordinate system whose origin is at the display center. O and R are computed as follows:

$$O = H / \tan e - D$$

$$R = O \sin a$$

Angles e and a can be easily calculated from the other labeled points in the figure. The O and R values are then used to set the initial user scale and position.

Note that while this start position balances the four goals, the user can still move his head to violate these goals. Only dynamic placement of the planet can prevent this. However, to avoid confusing the user we do not use dynamic placement of the initial position.

5.2. Zoom

To navigate, the user activates one of 3 modes by separate buttons on the laser pointer. These modes are zoom, pan, and rotate. As the user employs one of these modes, there is also an automatic viewing adjustment step which keeps the terrain stereoscopically viewable. This adjustment step moves the viewer along a direction perpendicular to the projection plane to keep terrain from appearing too far above or too far below the display surface (see section 5.5).

The first mode is for zooming. The zooming technique involves a user-controlled scale and translation plus an automatic rotation. The scale and translation

work as follows. When the user presses the zoom button, the current pointer position is recorded. As the button is held and the pointer is moved towards or away from the projection plane, the magnitude of the displacement from the initial position is computed. The magnitude determines the zoom speed. The direction of zoom, either in or out, is determined by whether the pointer is displaced closer to or farther from the projection plane. To perform the zoom, we first scale the platform coordinate system up or down based on magnitude and direction of the pointer movement. This has the effect of changing the physical-world to virtual-world scale factor, making the perceived world shrink or grow. Next, if the pointer intersects the terrain, the platform coordinate system origin is simultaneously scaled about this intersection point. This causes the user to zoom about the selected point. This technique gives the user control of zoom speed and direction plus control of the zoom-in point. In practice, users have found this method easy to use.

Since giving the user control of the zoom-in location gives her two more degrees of freedom to manage, we initially considered simpler approaches. For example, we tried always zooming about a fixed point such as the display center. However, we informally observed that this leads to repeated switching between zooming and panning while moving to a target. A user aligns her target with the display center by panning and then starts to zoom. Soon she is zooming slightly off target, which requires a correcting pan. This occurs repeatedly while approaching a single target and soon becomes a nuisance. Giving the user control of the zoom point avoids this problem.

In addition to this scale and translation activity, the user is automatically repositioned so that the planet appears to smoothly rotate about the selected terrain point. In detail, the planet rotates so that the planet normal vector at the selected terrain point becomes perpendicular to the projection plane (Figure 4b). Without this automatic rotation, a zoom quickly brings too much of the planet out of the projection plane leading to image fusion problems and severe frame cancellation (Figure 4a). The automatic rotation also keeps the viewing adjustment step, discussed in Sec. 5.5, from pushing the planet deeper into the display plane and driving the target location farther away.

There are a few additional automatic activities and rules applied when zooming. First we only activate the adjustment step when the world scale is past the threshold at which the planet diameter is 2.5 times the display width. This prevents the adjustment step from dynamically repositioning the carefully chosen start position. The next automatic activity guarantees a return to the initial position when zooming out. When the planet diameter is less than 2.5 times the display width and the user is zooming out, we smoothly move the user so as to slide the planet back to its starting position while

maintaining its current orientation. The repositioning occurs as a function of the difference between the user's current scale and the initial scale. This guarantees the initial scale and initial position are reached simultaneously.

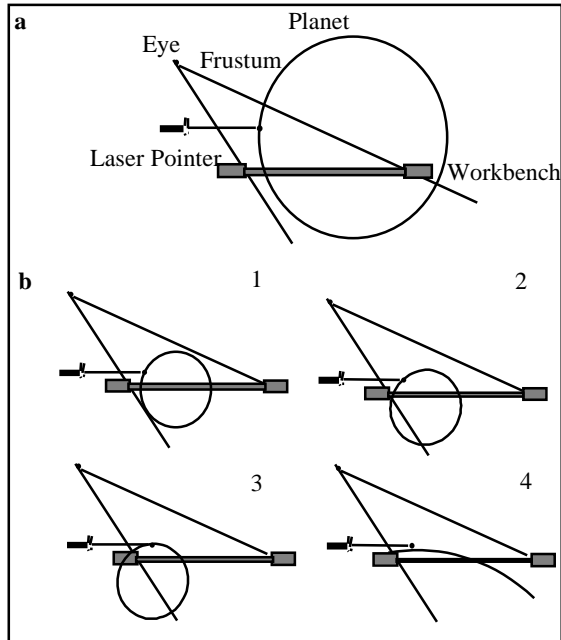


Figure 4: Zoom problem (a) and solution (b)

5.3. Panning

The next navigation mode is panning. The user grabs a point on the terrain with the laser ray and then drags that point around at the end of the ray. This is motivated by the results of Johnson for panning 2D scenes in a touch-controlled display [13]. Briefly, our panning method is analogous to Johnson's 'Background panning' method, also known as the 'click and drag' method, which he found to provide the best accuracy and to be the method most users expected. Note that our panning method differs from [7] where the grabbed terrain point is always the point directly underneath the 6-DOF device. Since a user can easily point our pointing device straight down, our method subsumes the method used in [7].

In more detail, the panning method works as follows. When the user presses the pointer's pan button, the point on the terrain intersected by the virtual laser is recorded. As the user moves the laser pointer, we track the intersection of the laser with a planet-centered sphere which intersects the recorded point. We then move the user to effectively rotate the planet about its center and bring the terrain point to the sphere intersection point. This panning method is effective at all zoom levels from those at which the planet appears as a small globe to those where the terrain appears planar. Also note that our

ray-terrain intersection code [31] allows for intersection with terrain outside the view frustum. This adds a bit more flexibility since the user can grab unseen terrain and therefore make larger panning gestures than would be possible if only on-screen terrain could be selected.

5.4. Rotation

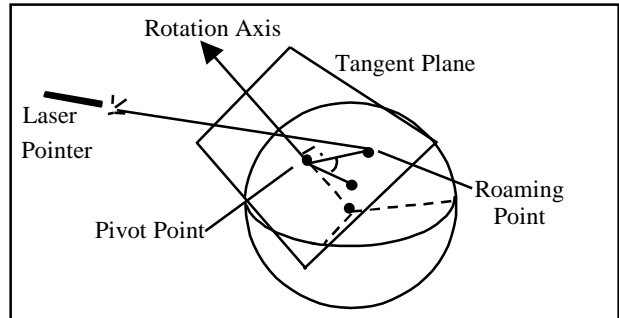


Figure 5: Rotation geometry

The final mode is rotation. When the user presses the rotation button the point on the terrain intersected by the virtual laser is recorded. This point is called the pivot point. Next the plane which intersects the pivot point and is tangent to the planet's spheroid is also calculated. While the button is held, the intersection point of the virtual laser and this plane is computed. This point is called the roaming point. As the user moves the roaming point, a thick line between the pivot point and roaming point is displayed. This thick line is rendered as a red transparent cylinder with a diameter set to be 5% of the display width. We call this cylinder the "rotation cue". While the user moves the roaming point about the pivot point, the planet rotates about the axis defined by the tangent plane's normal and the pivot point. Figure 5 illustrates these details.

It also was necessary to implement a small dead zone around the pivot point. Unless the user draws the roaming point beyond a threshold distance from the pivot point, the planet is not rotated. Without this dead zone, the user tends to create unintended large rotations due to small hand motions, especially when initiating the rotation maneuver. The dead zone's threshold distance is 10% of the display width.

Finally during rotation the pivot point is sometimes offset from the initial terrain intersection point along the spheroid normal. This becomes necessary when, at the current scale, the terrain undulations appear more than a few centimeters high. Here the terrain can obscure the rotation cue especially if the pivot point occurs at a low elevation. To implement this feature, we use some information recorded by the adjustment step. Specifically, we use the highest visible terrain peak measured perpendicular to the projection plane. We simply offset the pivot point along the spheroid normal by a distance equal to difference between the high-peak

height and the pivot point height. Ideally we should use the distances measured perpendicular to the plane tangent to the spheroid; however, this adds computational expense. Instead, we only perform this offset step if the spheroid normal is within 5 degrees of the projection plane normal. This works well since, due to the zoom-in auto-rotation feature, whenever we reach scales at which terrain undulations can obscure the rotation cylinder, the visible portion of the planet is fairly flush with the projection plane. Finally, we also implemented a two handed version of this rotation using two virtual laser pointers. One pointer defines the pivot point while the other defines the roaming point. Both methods are suitable for all scale levels.

5.5. Adjustment step

Having covered the user controlled navigation activities, we now discuss the automatic adjustment step for maintaining good stereoscopic imagery. Our goal is to take maximum advantage of stereoscopic depth cues while minimizing diplopia, frame cancellation and image distortions.

While others working with non-head-tracked stereo vary modeled eye separation [29], we do not do this. The primary reason is that false-eye separation modeling in head-tracked stereo induces a distortion with a head-position dependent shearing component [30]. This causes the viewed scene to shear back and forth with head movement despite otherwise perfect display calibration. Additionally, the induced distortion will cause the hand-held six DOF devices and their virtual representations to be miss-aligned. Poor correspondence between hand held six DOF devices and stereo imagery is problematic [23][6][18]. Finally in many command-and-control applications, users demand accurately portrayed height data which can be warped by false-eye modeling [30]. For these reasons we use a true eye separation. Therefore to maximize stereoscopic depth cues, we aim to keep the terrain within 1.5 meters of the user, a distance where stereo is strongest as a depth cue [5]. For the virtual workbench, this means keeping the terrain as close as possible, while considering image fusibility and frame cancellation. Also keeping the terrain slightly above the display plane puts most objects within arms reach and lets the user contact objects which are stereoscopically above the display.

The adjustment step works as follows. The terrain rendering thread renders display lists which are created by various other threads such as the terrain renderer and object renderer. For the adjustment step, this thread copies a sample of the right-eye depth buffer generated from the terrain display list and then the navigation thread examines this copy. The navigation thread scans the depth buffer copy and finds both the farthest point above the projection plane and the point nearest to the eye. During the same loop, we also record the number of pixels, A , above the projection plane and the number of pixels, P , not equal to depth buffer clear-screen value.

Two rules are then applied. First if A/P is less than a threshold (85%), we move the user along the projection plane normal in order to bring the near point to a predetermined target height, TH , above the display plane. Second if A/P is greater than threshold, then we move the user along the opposite direction so that the far point is flush with the display plane.

While the first rule simply draws the terrain peaks out of display, the second rule counters the following problem. At certain scales and terrain formations, the first rule can bring an unnecessary amount of the terrain above the display plane. For example, a particular data set might contain a few peaks and then mostly flat land. At certain scales the first rule would cause all of the flat terrain to be floating above the display. Effectively there is a large plane which extends far beyond the window limits hovering above the display. Even at target heights as small as 5 centimeters, the uniformity and extent of this plane creates a strong frame cancellation effect. In contrast for the same target height, if the terrain is more undulating, then the frame cancellation effect is less disturbing. We surmise this occurs since with undulating terrain only some of the terrain at the display edges is clipped by the view frustum, while with the flat planar terrain *all* terrain at the display edge is clipped. Given this situation, the more natural position for this problematic terrain is with the planar area flush with the display plane. The second rule catches such cases--where too much terrain is above the display plane--and pushes the terrain back down.

For the Target Height, TH , we use a constant value that empirically works well. While TH could be adjusted as a function of the nearest fusible image plane [24] this would cause the terrain to be pushed down into the display plane when the user leans down for a closer look. We informally observed that such behavior is more unnatural than diplopic conditions. While people experience real-world diplopia when peering too closely to an object, they do not experience inanimate objects autonomously moving away when closely examined. Therefore TH is set to 10% of the standard user height above the workbench, a position within Yeh's fusibility constraints for the standard eye height (Figure 3).

Note that depending on user activities, object depth information from the z-buffer should be also considered in addition to terrain depth information. If the user wants to see aircraft, for instance, it is important to account for their depth values so that all aircraft are fusible and visible. On the other hand, if the user is focusing on the terrain, accounting for the aircraft could be problematic as it could push the terrain far below the display plane when bringing the aircraft into view.

Finally, while Ware effectively moves the near point instantaneously to the screen, we move the user at a logarithmic rate towards the target position. Using logarithmic transitions keeps our rule set from causing abrupt displacements.

6. Summary and future work

We have presented a third-person navigation interface that effectively balances the issues of scaling and stereoscopic display with head-tracking for a very large dataset. This navigation interface has been demonstrated to several dozens of technical and non-technical users (Figure 6). Several cycles of development and informal observations at these demonstrations led to the present design. Generally, users respond quite favorably and appear quite comfortable and adept after brief instruction and a few minutes of navigating.

Future work includes:

- development and integration of a first-person navigation technique and a suitable adjustment step
- formal individual testing of first person and third person navigation techniques
- formal comparative testing of first person navigation and third person navigation.
- formal evaluation of resulting navigation methods embedded in a simple command-and-control application.

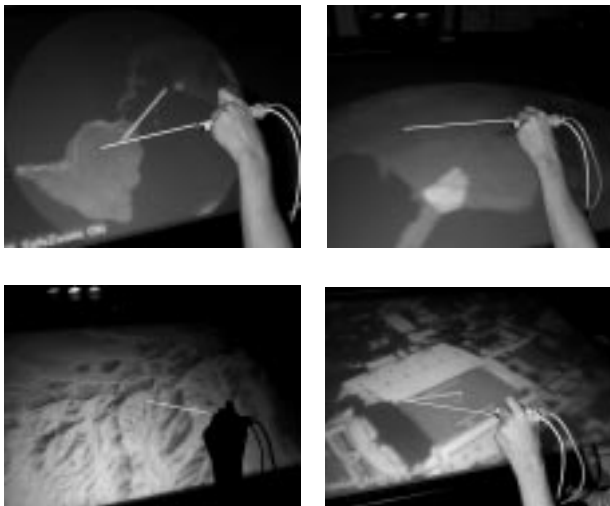


Figure 6: Top: Rotate, Pan; Bottom: Zoom, Rotate

7. Acknowledgements

This work was performed in part under contracts N00014-97-1-0882 and N00014-97-1-0357 from the Office of Naval Research. Support was also provided under contract DAKF11-91-D-004-0034 from the U.S. Army Research Laboratory. We thank Frank Jiang for help in setting up the workbench environment and creating images for this paper.

8. References

- [1] Barfield, Woodrow, Blake Hannaford, John M. Hollerbach, Thomas B. Sheridan, David Zeltzer, Michael Zyda. Presence: Teleoperators and Virtual Environments. April, 1998.
- [2] Bartram, Lyn., Albert Ho, John Dill, Frane Henigman. The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces. In Proceedings of ACM UIST'95, (1995, Pittsburgh, PA), ACM Press, pp.207-215.
- [3] Bederson, Benjamin B., James D. Hollan. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In Proceedings of ACM UIST'94, (1994, Marina Del Ray, CA), ACM Press, pp.15-16.
- [4] Bowman, Doug A. David Koller. Larry F. Hodges. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In Proceedings of IEEE 1997 Virtual Reality Annual International Symposium. pp. 45-52.
- [5] Cutting, James E. How the eye measures reality and virtual reality. *High-Performance Computing and Human Vision I*. Behavioral Research Methods, Instruments & Computers, 1997, 29(1), 27-36.
- [6] Deering, Michael. High Resolution Virtual Reality. *Computer Graphics*, 26, 2, July, 1992. pp. 195-202.
- [7] Durbin, Jim, J. Edward Swan II, Brad Colbert, John Crowe, Rob King, Tony King, Chris Scannell, Zachary Wartell, Terry Welsh, "Battlefield Visualization on the Responsive Workbench", Proceedings IEEE Visualization '98, October 18-23, Research Triangle Park, North Carolina: IEEE Computer Society Press, 1998, pp. 463-466.
- [8] Furnas, George W., Benjamin Bederson. Space-scale Diagrams: Understanding Multiscale Interfaces. In Proceedings CHI'95 Mosaic of Creativity, (1995, Denver, Colorado), pp. 234-240.
- [9] Fröhlich, Bernd. Berthold Krish. Wolfgang Krüger. Gerold Wesche. Further Development of the Responsive Workbench. *Virtual Environments '95. Selected Papers of the Eurographics Workshops*. pp. 237-246.
- [10] Goble, John C, Ken Hinckley, Randy Pausch, John W. Snell, Neal F. Kassel. Two-Handed Spatial Interface Tools for Neurosurgical Planning. *IEEE Computer*, July 1995.
- [11] Guan, Chua Gim. Luis Serra. Ralf A. Kockro. Ng Hern. Wieslaw L. Nowinski. Chumpon Chan. Volume-Based Tumor Neurosurgery Planning in the Virtual Workbench. In Proceedings of IEEE Virtual Reality Annual International Symposium. (Atlanta, GA. 14-18, March 1998). pp. 167-173.
- [12] Hix, Deborah. James N. Templeman. Robert J.K. Jacob. Pre-Screen Projection: From Concept to Testing of a New Interaction Technique. In Proceedings of CHI'95 Mosaic of Creativity, (1995 Denver, Colorado). pp. 226-232.
- [13] Johnson, Jeff. A. A Comparison of User Interfaces for Panning on a Touch-Controlled Display. In Proceedings of CHI'95 Mosaic of Creativity, (1995 Denver, Colorado). pp. 218-225.
- [14] Krüger, Wolfgang. Christian-A. Bohn, Bernd Fröhlich, Henrich Schüth, Wolfgang Strauss, Gerold Wesche. The Responsive Workbench: A Virtual Work Environment. *IEEE Computer*, vol. 28. No. 7. July 1995. pp 42-48.
- [15] Lieberman, Henry. Power of Ten Thousand: Navigating in Large Information Spaces. In Proceedings of ACM

- UIST'94, (1994, Marina Del Ray, CA), ACM Press, pp.15-16.
- [16] Leigh, J., A.E. Johnson, C.A. Vasilakis, T.A DeFanti. Multi-perspective collaborative design in persistent networked virtual. Proceedings of the IEEE 1996 Virtual Reality Annual International, Santa Clara, CA, USA 30 March-3 April 1996. pp.253-60, 271-2. 1996.
 - [17] Lindstrom, Peter, David Koller, William Ribarsky, Larry Hodges, Nick Faust, and Gregory Turner. Real-Time Continuous Level of Detail Rendering of Height Fields. *Computer Graphics (SIGGRAPH 96)*, pp. 109-118 (1996).
 - [18] McKenna, Michael. Interactive Viewpoint Control and Three-Dimensional Operations. ACM 1992 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics, pp. 53-56.
 - [19] Mine, Mark. R., Frederick P. Brooks Jr., Carlo H. Sequin. Moving Objects In Space: Exploiting Proprioception In Virtual Environment Interaction. *Computer Graphics Proceedings, Annual Conference Series*, pp.19-26 (1997).
 - [20] Obeysekare, Upul, Chas Williams, Jim Durbin, Larry Rosenblum, Robert Rosenberg, Frenando Grinstein, Ravi Ramamurti, Alexandra Landsberg, William Sandberg. Proceedings IEEE Visualization '96. pp. 345-9, 500.
 - [21] Rosenblum, Larry (editor). Applications of the Responsive Workbench. *IEEE Computer Graphics and Applications*, vol. 17, no. 4. July-August 1997. Pp. 10-15.
 - [22] Serra, Luis. Tim Poston. Ng Hern. Heng Pheng Ann. Chua Beng Choon. Virtual Space Editing of Tagged MRI Heart Data. *Computer Vision, Virtual Reality and Robotics in Medicine. First International Conference, CRMEd '95.* (1995). pp. 70-76.
 - [23] Schmandt, Christopher. Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Workstation. *Computer Graphics*, 17, 3, July 1983.
 - [24] Southard, David A. Viewing Model for Virtual Environment Displays. *Journal of Electronic Imaging* 4(4), pp 413-420. 1995.
 - [25] Stoakley, Richard, Matthew J. Conway, Randy Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of CHI'95 Mosaic of Creativity*, (1995 Denver, Colorado). pp. 266-272.
 - [26] Veron, Harry. David A. Southard. Jeffrey R. Leger. John L. Conway. Stereoscopic Displays for Terrain Database Visualization. *SPIE: Stereoscopic Displays and Applications*, Vol. 1256, pp. 124-130. 1990.
 - [27] Yeh, Yei-Yu, Louis D. Silverstein. Limits of Fusion and Depth Judgement in Stereoscopic Color Displays. *Human Factors*, 1990, 32(1), 45-60.
 - [28] Ware, Colin. Dynamic Stereo Displays. In *Proceedings of CHI'95 Mosaic of Creativity*, (1995 Denver, Colorado). pp. 310-316.
 - [29] Ware, Colin. Cyril Gobrecht. Mark Paton. Algorithm for Dynamic Disparity Adjustment. *Proceedings of the SPIE - The International Society for Optical Engineering. Stereoscopic Displays and Virtual Reality Systems II.* San Jose, CA, USA. 7-9 Feb. 1995. vol.2409 pp.150-6
 - [30] Wartell, Zachary, Larry Hodges, William Ribarsky. The Analytic Distortion Induced by False-Eye Separation in Head-Tracked Stereoscopic Displays. *GVU Lab Technical Report*, no. 99-01, 1999.
 - [31] Wartell, Zachary, William Ribarsky Larry Hodges. Efficient Ray Intersection with Global Terrain using Spheroidal Height-Augmented Quadrees. *GVU Lab Technical Report*, no. 98-38, 1998.